**AZERBAIJAN STATE OIL AND INDUSTRY UNIVERSITY**

**Faculty: “Information Technologies and Control”**

**Department: “Computer Engineering”**

**Group: 672.7E**

**Specialty: 050655 – “İnformasiya texnologiyaları”**

**Graduation work**

**Frontend Development of online examination system**

Consultants:

Student: Abbasov Nihad Abdulla oğlu

Supervisor: assoc. prof. Hajiyev Y.M.

Compliance assessment: assoc. prof. Huseynova A.N.

Head of the department: assoc. prof. Ragimova N.A.

**BAKU – 2021**

# **Abstract**

**Subject:** “**Frontend Development of online examination system**”

**The research purpose:** The aim of this project is to investigate the need for an online assessment system and to create a web application to administer an examination process for students in order to exclude them from the whole course. The study on the problems and strategies for the online assessment method is also part of the programme.

**The research result:** The program definitely saves resources such as time, money, and human activity. To prevent many exam management processes such as creating queries, reviewing, and reporting results, the system's automation greatly decreases the examiner's job and the organization's expenses. The simulated environment relieves students of tension and reduces the likelihood of cheating during test administration.

**Keywords**: web development, frontend, JavaScript, Bootstrap, plugins

**Xülasə**

**Mövzu:** "**Onlayn imtahan sisteminin Frontend İnkişafı**".

**Tədqiqatın məqsədi**: Bu layihənin məqsədi onlayn qiymətləndirmə sisteminə ehtiyacın öyrənilməsi və tələbələrin bütün kursdan kənarda qalması üçün giriş imtahanını idarə etmək üçün veb tətbiqetmə yaratmaqdır. Onlayn qiymətləndirmə metodu ilə bağlı problemlər və strategiyalar üzərində işləmək də proqramın bir hissəsidir.

**Tədqiqatın nəticəsi:** Proqram qəti şəkildə vaxt, pul və insan fəaliyyəti kimi mənbələrə qənaət edir. Sorğu yaratmaq, nəticələrin nəzərdən keçirilməsi və hesabat verilməsi kimi bir çox imtahan rəhbərliyi ilə bağlı mürəkkəb proseslərin qarşısını almaq üçün sistemin avtomatlaşdırılması imtahan verən şəxsin işini və təşkilatın xərclərini xeyli azaldır. Təmin olunan süni mühit şagirdləri gərginlikdən azad edir və sınaq zamanı köçürmə və bu kimi digər halların ehtimalını azaldır.

**Açar sözlər:** vebin inkişafı, frontend, JavaScript, Bootstrap, plaqinlər

**Реферат**

**Тема**: «**Frontend разработка системы онлайн-экзаменов**».

**Цель исследования:** Целью этого проекта является исследование необходимости онлайн-системы оценивания и создание веб-приложения для администрирования процесса экзамена для студентов, чтобы исключить их из всего курса. Изучение проблем и стратегий для метода онлайн-оценки также является частью программы.

**Результат исследования:** Программа определенно экономит такие ресурсы, как время, деньги и человеческую активность. Чтобы предотвратить многие процессы управления экзаменом, такие как создание запросов, просмотр и отчетность о результатах, автоматизация системы значительно снижает работу экзаменатора и расходы организации. Смоделированная среда снимает напряжение с учащихся и снижает вероятность обмана во время проведения теста.

**Ключевые слова:** веб-разработка, интерфейс, JavaScript, Bootstrap, плагины

**CONTENTS**

**INTRODUCTION** [**6**](https://docs.google.com/document/d/1tuq5Ao9k3dLaBlJTD1iKLQGkJkEFxzIKFeGFner5-9I/edit#heading=h.30j0zll)

**CHAPTER 1. Research of a subject area 7**

1.1. Core of Web Development [**7**](https://docs.google.com/document/d/1tuq5Ao9k3dLaBlJTD1iKLQGkJkEFxzIKFeGFner5-9I/edit#heading=h.3znysh7)

1.2. Web protocols **10**

1.3.Fields of Web Development **12**

**CHAPTER 2. Principals of development tools used in the project 16**

2.1. Essentials of Front-end Development **16**

2.2. CSS frameworks and preprocessors **31**

2.3. JavaScript Workflow **27**

2.4. JavaScript most popular frameworks **31**

**CHAPTER 3. Construction of online examination portal 34**

**3.1**. Getting environment ready for the project **34**

**3.2.** Creating frontend folder and file structure **34**

**3.3.** Constructing Homepage **36**

**3.4.** Adding plugins to the project **39**

**CONCLUSION 40**

**REFERENCES 41**

**INTRODUCTION**

The utility of an online examination environment and the automation of assessing students by analyzing them in real-time are inextricably linked. Along with constantly changing teaching and learning techniques, procedures such as tests must also be scrutinized. The result of such an analysis is an online review process, which automates, digitizes, and flattens the system to make it more accessible, inclusive, and precise than its previous iteration.

**Chapter 1: RESEARCH OF A SUBJECT AREA**

This chapter provides information about fundamental knowledge about web development and how web works.

**Chapter 2: principals of development tools used in the project**

This chapter provides information about fundamentals of development tools used in the project.

**Chapter 3: CONSTRUCTION OF ONLINE EXAMINATION PORTAL**

In this last chapter, improvement of the building a project was shown and the user interface part was displayed.

**The scope of work**. The graduation work consists of 41 pages and 17 images. 22 literatures were used in this work. The structure of this work consists of an introduction, 3 chapters and a conclusion.

**CHAPTER 1. Research of a subject area**

* 1. **Core of Web Development**

Web creation is the process of creating a Web site for the Network. Online development can range from producing a single static website in plain text to developing complex online applications, electronic businesses, and social network services. Website development, User experience design, Web content construction, customer liaison, client-side/user-side scripting, Web database administration configuration management, and e-commerce advancement are some of the more comprehensive practices to which Web development typically relates. /1/

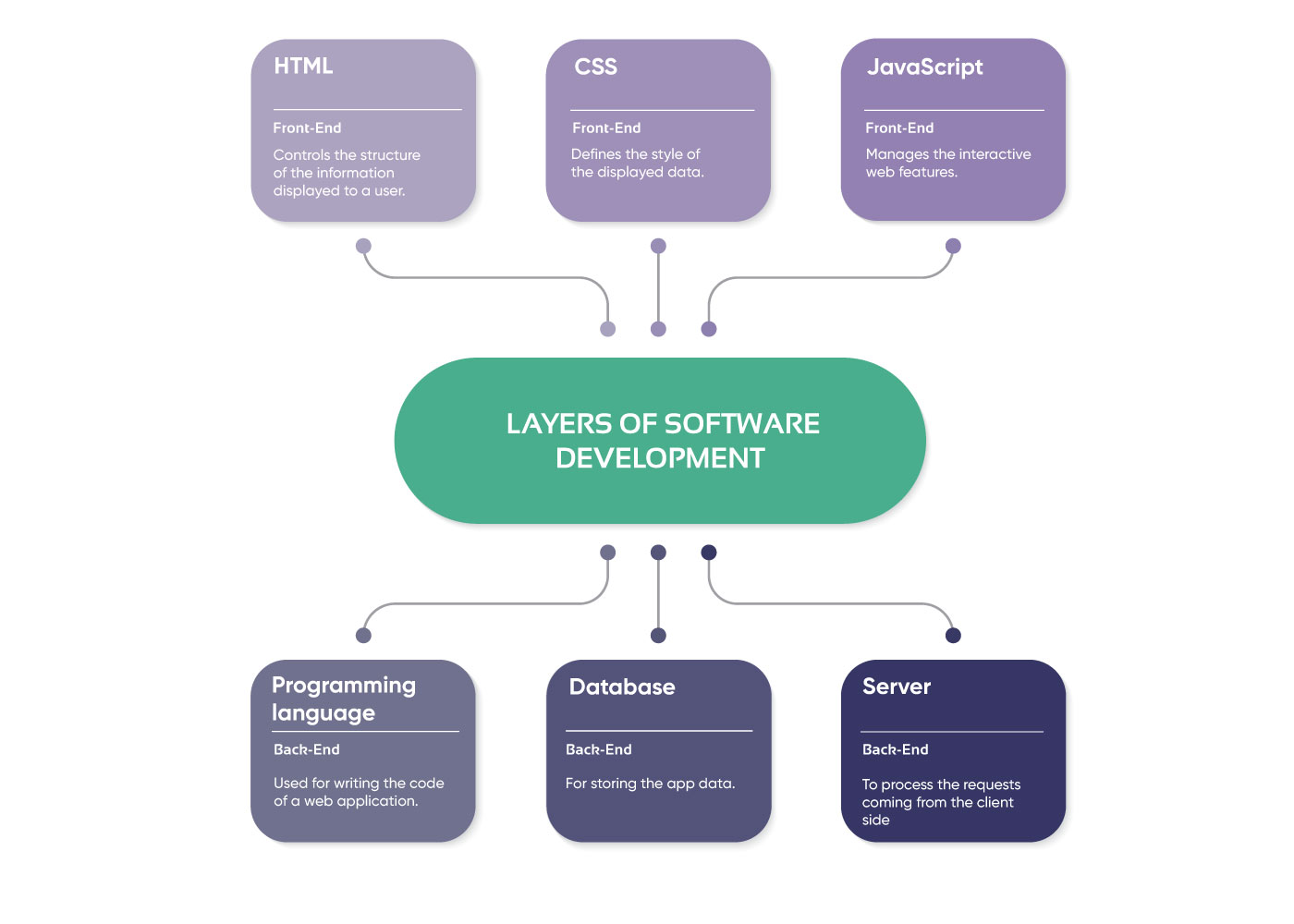


Figure 1.1 Divison of Web Development

"Web construction" is often used by Web practitioners to refer to the key non-design elements of creating Web sites: creating markup and programming. Data management frameworks can be used in web construction to make content updates faster and more accessible to those with specific technical knowledge. Web development is the process of developing and maintaining websites; it is the work that happens in the world to make a website looking attractive, act effectively, as well as provide a consistent level of service. Web developers, also described as "devs," accomplished this by leveraging a broad array of computer languages. The languages they choose are controlled by the duties they execute and the platforms they work on.

**What is Website?**

A website is a collection of interconnected websites that can be visited by a homepage or a browser such as Opera, Google Chrome, Internet Explorer or Mozilla.

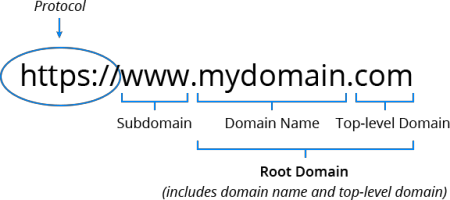


Figure 1.2 Custom domain name’s components

Every webpage has its own URL, which is a completely distributed address defined as a domain name. A URL is formed composed of the following components:

- The **protocol** used to access the website, which in this case is http, meaning port 80.

- The subdomain, which is www by default.

- The domain name; often, domain names are chosen for their significance.

- A suffix name, such as.com,.info,.net,.biz, or a country-specific suffix.

- The directory, or in simple language, a folder on the server where this website is stored. /2/

**How does a web browsers work?**

A web browser allows you to access any website on the internet. It gathers data from many places of the internet and displays it on your desktop or mobile device. The data is sent over the internet via the Hypertext Transfer Protocol, which specifies how text, pictures, and video are delivered. This information must be shared and displayed in a uniform way so that it can be viewed by users using any browser, anywhere in the globe.

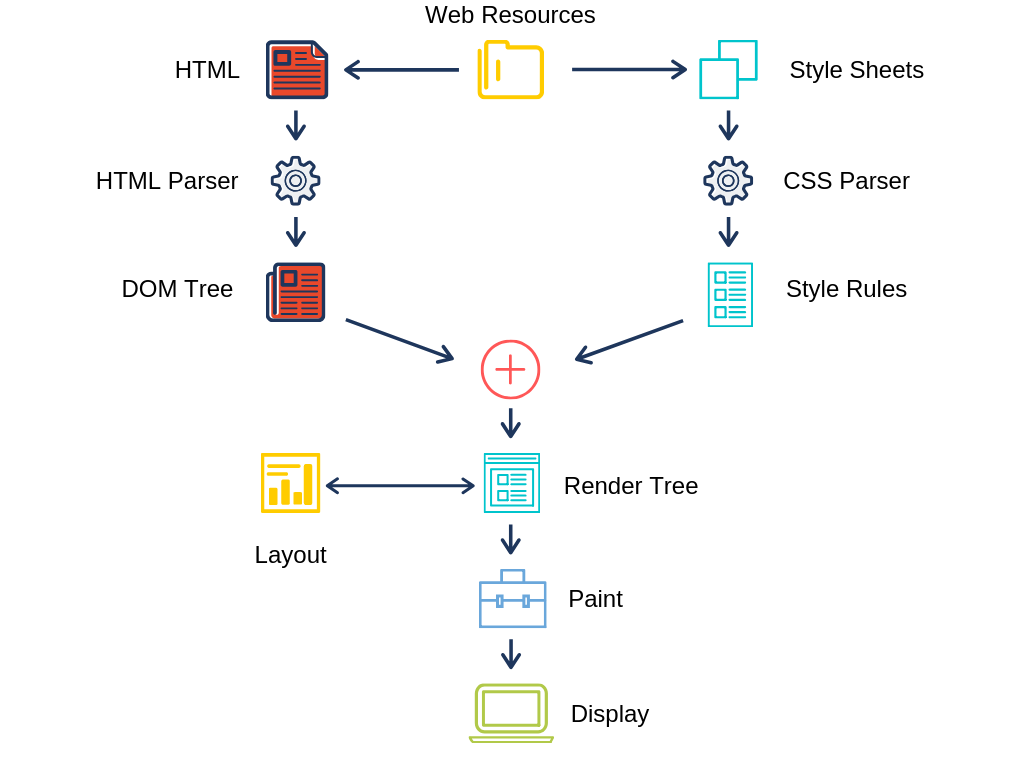


Figure 1.3 Web workflow

Regrettably, not all browser developers read the format in the same way. This means that a website might have a varied look and feel for users. The term "web standards" refers to the process of establishing consistency between browsers so that any user, regardless of browser, can use the internet. When a web browser requests information from an internet-connected server, a rendering engine converts the information into text and pictures. Session management decodes codes that belongs to HTML in order to construct what we see and experience on the internet.

Users can utilize hyperlinks to navigate to other online pages or sites. Thus every webpage, photo, or videos does have its own Uniform Resource Locator (URL), typically known as a web identity. When a browser receives information from the database, the web address directs the browser on where to look for each item described in the html, which then informs the browser on where the item will display on the web page.

**Cookies**

Cookie files can be used by internet sites to just save data about you. Cookies remain cached on your device which will be accessed the next time you visit that site. The webpage code will analyze that file whenever you return to authenticate that it really is you. A cookie, for instance, permits a website to remember your login credentials when you revisit it. Cookies can also keep track of more specific information about you. Perhaps your hobbies, web browsing habits, and so on. This implies that a website can give you with more relevant material - usually in the shape of advertisements. There seem to be cookies which it emanate from sites you're not even visiting at the moment and therefore can monitor you from place to place to collect data about users, which is periodically sold to many other corporations. These cookies are known as third-party cookies. Although not all browsers enable it, you can sometimes block these types of cookies.

* 1. **Web protocols**

Several technological advances have leveraged its performance on the Internet for the last few centuries. Nonetheless, the term Internet has become much more widespread in view of the rapid of a massive growth in research, increasing productivity, and application development. Generally, UDP has been used as a transport layer protocol for real-time applications because this is a more simpler protocol unlike TCP because it wouldn't require connection setup latency, flow control, or retransmission. TCP-based applications, such as the Hypertext Transfer Protocol (HTTP) and Hypertext Transfer Protocol Secure (HTTPS), usually require more than 80% of the Internet's bandwidth (HTTPS). This uses a sliding window flow mechanism under TCP flow control. Packet loss can then be used to determine network traffic. When this occur, however, the packet is reprocessed. TCP is still generally speculated to be the dominating protocol of Internet traffic.

**What is HTTP?**

HTTP is the acronym for the Hypertext Transfer Protocol through its full form. HTTP seems to be a set of rules and standards that define how information can be sent throughout the Web. HTTP specifies technical standards for browser and the web server. HTTP is a network protocol which runs in the background of TCP just at application layer. HTTP utilizes hypertext structured text to form a logical link between text-containing nodes. It's characterized as that of the "stateless protocol" since every command is executed separately, without using the previous processed command as a reference.

**What is HTTPS?**

HTTPS is a highly advanced and secure HTTP version. It means Hyper Text Transfer Protocol Secure. For data communication, it uses port 443. By encrypting all communication using SSL, it provides for secure transactions. It's a hybrid of the SSL/TLS and HTTP protocols. It enables the network service to somehow be recognised inside of an encrypted and sensible manner. HTTP also helps the servers and search engine to maintain a secure channel. It provides data security in both directions. This assists you in preventing the theft of potentially sensitive information.

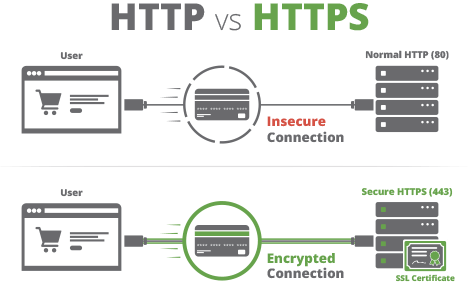


Figure 1.4 HTTP vs HTTPS

**HTTP advantages:**

HTTP has the following advantages:

* HTTP pages are bookmarked on technology and the internet, making them extremely available.
* Platform is flexible, allowing for cross-platform transfer.
* No Runtime support is required.
* Can be used over firewalls. It's indeed suitable to implement global programs.

**HTTPS benefits:**

* The most with HTTPS-enabled webpages can have a reroute in setup. As a nutshell, even though users input HTTP://, you'll immediately forwarded to https over such a secure network.
* It enables users to conduct secure online transactions, such as banking.
* SSL technology secures and promotes confidence among all users.
  1. **Fields of Web Development**

The construction of creating dynamic web pages is conducted using web programming, generally termed as web development. Web applications encompass social media platforms along with Facebook as well as e-commerce platforms such as Amazon. Finding a group of technologies that are specially suitable for both client-side and server-side components of your web project is the key to effective development. A programming language seems to be a right model composed of a set of characters which really produces different types of bytecode. In computer programming, programming languages are used to implement algorithms. The majority of programming languages are made up of computer instructions. Rather than using the standard scripting languages, reconfigurable systems choose a package of customized algorithms. Web development is separated into two sections: **frontend** and **backend**. This saying sometimes known as client-side and server-side development.

**Client-side Development**

The user interface of a program seems to be the part that consumers can see constitutes the platform's "face" or "facade." As a consequence, it's almost always refereed to this as the "front-end". There are not too many digital stacking components affiliated with just this programming stack:

* The architecture of something like the data displayed to just a consumer in a browser window is managed by that of the **Hypertext Markup Language (HTML).**
* **Cascading Style Sheets (CSS)** regulate characteristics also including fonts, styles, content and background palettes to form the appearance of said data extracted.
* **JavaScript (JS)** seems to be in handle of its application's interactive web functionalities.

With in service of trained specialists, most of these systems are effective resources. Their union, on the other hand, opens up a whole new world of opportunity for web application developers. Whilst JavaScript, HTML, and CSS are recognized industrial guidelines, they can also be replaced with software applications like Apache Flex as well as others.Through use of frameworks is indeed a popular method which accelerate the application development. Frameworks and libraries for front-end development encompass:

* **Angular** is a Google-developed tool for building flexible configuration files for application development.
* **React** is just a Facebook Inc. toolkit for constructing single-page programs, essentially signifies that even an interface could exhibit relevant metrics on the very same webpage without having to reload it. Engineers might quickly design extensible interactive solutions leveraging JSX, React's syntactic extension from Javascript.
* **Vue.js** — a compact JavaScript framework for constructing adjustable web applications for single-page web services that really has gotten a lot of attention between many programmers that since debut in 2014. Modern front-end development for web apps entails a mixture of various modules: libraries and frameworks, as a single one is no longer practical to fulfil a customers ’ expectations. As either a response, dedicated professional web softwares are now becoming extremely prevalent. The foregoing are just some of those widely recent features:
* Bootstrap — a package of JavaScript and CSS templates for generating web - based applications. The latter's principal objective is to improve accessibility as well as assist the “mobile-first” digital design paradigm. It's worth noting that this framework is mostly focused on web pages rather than internet platforms.
* Foundation Is indeed a combination of specialized functional frameworks for webpages, services, and e-mails. The integrated set of useful HTML, JS, and CSS modules targeted on responsiveness and a “mobile-first” strategy saves so much time for front-end professionals.

**Server-side Development**

The back end, or server side, of a web service is concealed from viewers and incorporates every one of the pieces necessary to deliver the application, similar to the compounds present beneath a window and the inside of the bone structure of an automobile.

The preceding are server-side aspects:

* A scripting language when constructing software product programs. There seem to be numerous platforms and packages helping facilitate and optimize the data collection, and there are also on client-side programming. Many of the most common backend scripting languages, such as Java, PHP, Ruby, Python, and Scala, are actually related methodologies: Django, Laravel, Ruby on Rails, Spring, and Play, respectively.
* The app's data is saved in a database. Owing to the database management paradigm, those assets may also be relational or non-relational. Non-relational (non-SQL) platforms implement different versions regarding storing than relational database systems. Oracle, MySQL, Microsoft SQL Server, PostgreSQL, and MongoDB are also the five most important systems based on the current DB-Engines portal. However, about the last one, those who have all been linked. Nonetheless, inside the world of web software design, priorities vary slightly, and users ought remain cognizant of something like this.

**CHAPTER 2. PRINCIPALS OF DEVELOPMENT TOOLS USED IN THE PROJECT**

**2.1**.  **Essentials of Front-end Development**

**Basics of HTML**

HTML (HyperText Markup Language) is a current standard markup language that employs common acronyms called "tags" to tell a web browser how the author wants sections of a web page to be laid out. It should be the basis through which all Web sites are constructed. HTML seems to be a sign language that characterizes Search engines that used a subset standard frame tags. It is not a programming language. Tag names are enclosed in angle brackets, and mark-up is made up of tags.

**History of HTML**

It was created in March 1989 by British scientist Tim Berners-Lee at some of the main companies which seems to be utilize at Switzerland to allow business physicists to exchange all computer-stored information. Berners-Lee developed a text browser that uses hypertext to offer point-and-click navigation over the internet. In May 1990, this system was dubbed the World Wide Web, and in 1993, college student Marc Andreessen improved it by adding an image tag. The World Wide Web exploded in popularity after HTML was able to show both text and images. As web browsers evolved, their creators began to add proprietary tags of their own, thereby creating their own versions of HTML! The World Wide Web Consortium (W3C) recognized the risk of HTML fragmentation and produced a common specification that all web browsers should follow. The browser makers were successfully persuaded to support the standard tags as a result of this. Until the publication of HTMLversion 4.01 at the turn of the century, the W3C's HTML standard was constantly amended to provide new capabilities. The W3C also issued an XHTML (eXtensible HTML) definition at the time, which required all code to be "well-formed" in order to conform with the standards of the eXtensible Markup Language (XML). Berners-Lee confesses that his efforts to force web authors to use strict syntax did not succeed. So, in October 2014, the W3C returned to HTML and issued a definition for version 5. On November 1, 2016, it was updated to version 5.1. The supported capabilities of HTML5.1 are defined and demonstrated in this book, which is referred to as “HTML5” or just “HTML” in general.

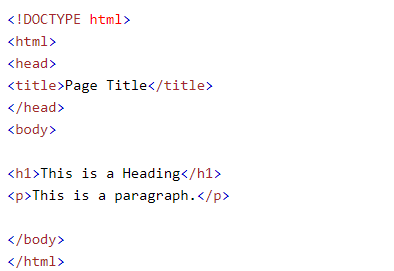


Figure 2.1 Ordinary code layout in HTML

**Addressing Web Pages**

The World Wide Web is viewed as a collection of high-capacity networked computers "web servers" that are attached to the network via telephone service and satellites. The HyperText Transfer Protocol (HTTP) is a data transmission service that enables each and every device associated to something like a remote server to sync bookmarks throughout the online. HTML website content along with index.html, are typically files that contain preserved with either the “.htm” or “.html” executable file.

**Defining document structure**

The three parts of an HTML5 document's structure are as follows:

* *Declaration of document type* – specifies which version of HTML is used to mark up the document.
* The *head section* contains descriptive information about the document, such as the title and character set used.
* The material that will show when the document is loaded into a web browser is contained in the *body section.*

**<!DOCTYPE HTML>**

**<!DOCTYPE html>**

**<!Doctype Html>**

**<!doctype html>**

Every HTML5 document must have a document type declaration at the beginning of the first line in order for the web browser to “render” (display) the document in “Standards Mode” - as defined by the HTML5 specifications. Because HTML5 is not a case-sensitive language, the document type declaration tag, as well as all other tags, can be written in any combination of uppercase and lowercase characters.

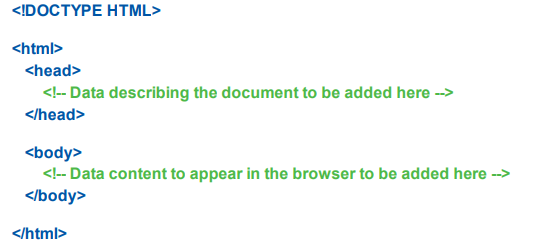
****

Figure 2.2 HTML file model

**Head section**

The head section of a document starts with an HTMLopening tag and finishes with an HTMLclosing tag. To fulfill the HTML article's head element, content characterizing the item can be released later between such two tags.

**Body section**

This section of the manuscript commences with either an HTML opening element and culminates including an HTML closing tag. To finalize the HTML document's body section, relevant data to displayed in the reader can be announced later between these two tags. /17/

**Code comments**

Comments can be put between tags at any point in both the head and body sections. The browser dismisses everything which happens between both the commenting tags.

**Creating a .html document**

Any plain text editor, such as Windows' Notepad, can be used to produce a simple HTML5 page using the core HTML5 document structure. To generate a valid "barebones" HTML5 document, information describing the document's primary textual language, character encoding type, and title must be added first. A declarative programming value is allocated to a lang "attribute" inside this opening <html> parent element to represent the document's main language. The syntax for English is en, furthermore the entire opening parent item appears such as this one:

<html lang=“en”>. /3/

**Describing the Document**

In addition to describing the document's character setting and safety measures, *<meta>* tags can then be used to convey descriptive data to search engines. However, this does not ensure a good position, as search engines also use other page information, particularly the document title, for this reason. Nonetheless, it is beneficial to provide a description and a list of keywords relevant to the page's contents so that search engine "spiders" can index the page. A name attribute is always used to indicate a page feature, and a content attribute is used to define the value of that feature. The word "description," for example, allows you to specify text content that describes the page. Short, succinct sentences that could appear in a search engine's results page should be used. Any description that is more than 200 characters long may be shortened. /5/

**Incorporating scripts**

Scripts would be used to interact with the consumer and deliver possible influence in Http response. With the growth of Web 2.0 pages, where elements of the page can be dynamically updated, this feature has become increasingly crucial. Previously, the browser would request a full new page from the web server, which was inefficient and inconvenient, therefore Web 2.0 represents a huge step forward. Scripts surrounded by the script> /script> tags can be included in the head section of an HTML document, but they are best kept separate to comply with HTML5's goal of separating content from display. Because scripts are expected to use the JavaScript language by default, the type attribute value of the <script> tag is automatically set to “text/javascript.” This means that unless you're including a script that uses a different scripting language, the type attribute can be omitted from the tag. When using an external script, the src property in the <script> element must be set to the URL of the script file. **<script src=“script.js”> </script> /**4**/**

**Incorporating style sheets**

To customize the communicative properties of every other object on the webpage, style sheets can indeed be implemented throughout HTML projects. All parts of HTML that were previously associated to display have really been overtaken with the use of style sheets. The <font> tag, for instance, is therefore no longer wanted even though font family, weight, style, and size now are supplied by such a style sheet regulation. In the head portion of an HTML document, style sheets enclosed by <style> </style> tags can be used to include rules defining how the content will appear. Because style sheets are assumed to use the Cascading Style Sheet language by default, the type attribute value of the <style> tag is automatically set to “text/css.” This means that unless you're using a style sheet that uses a different styling language, the type attribute can be omitted from the element. A rudimentary style guide comprising instructions to establishing the appearances of any and all size-one headlines, for instance, might looks like that really:

**<style>**

**h1 {**

**color : red;**

**background : yellow;**

**}**

**</style>**

**What is CSS?**

The language that is being used to style a Html page is CSS. It dictates the way Html tags would appear. When it comes to how HTML and CSS should work together, the W3C's attitude is to utilize HTML components to specify a web page's content and CSS to determine its appearance. It is a set of rules that allows the user to control the appearance of a web content in a web browser. Colors, background pictures, typefaces (fonts), margins, and indentation are all examples of formatting applied to a website. format (a range with configuration options such styles, dimensions, as well as hues) and thereafter implement that one to on or maybe more elements of something like the manuscript an HTML page using a selector. Several CSS features could be used to adjust the look of a site. CSS works by applying rules to elements within a web page. Selectors are used by browsers to decide which items to apply CSS rules to. Tags for structuring a web page were never intended in HTML! HTML was established to define a web page's content, such as:

<h1>It is the heading</h1>

<p>It is the paragraph</p>

When features such as <type> as well as hue characteristics being implemented into the HTML specification, scripters were approved to a tail spin. The creation of large websites, now types and hue statistics were added to every page, remains a time saver as well as costly process. CSS was introduced by that of the World Wide Web Consortium (W3C) to deal with this issue. The HTML page's style structure was modified by CSS! /6/

**What is JavaScript?**

The client-aspect scripting language JavaScript is most often employed. This indicates that JavaScript code is embedded in an HTML page. When a devotee requests a hypertext markup language website with JavaScript, the script is sent to the browser, which then instructs it to sanction to one matter. JavaScript is an interpreted programming language that is high-level, dynamic, and untyped. The ECMAScript language specification includes it as a standard. That's one of the three technological advances enabling constructing internet content, together HTML and CSS; which is used by the overwhelming bulk of publications and therefore is recognized among all modern operating systems without a need for plug-ins. JavaScript is now a multi-paradigm system that accommodates object-oriented, impulsive, and reactive programming styles, thanks to it's own technology demonstrator layout and first-class functionalities. /7/

**JavaScript history**

Tim Berners-Lee created the original version of HTML between 1989 and 1991, and it was quite static. Web pages just displayed content, with the exception of link jumps made with the an element. Brendan Eich, a Netscape employee, thought it might be useful to add dynamic functionality to online sites in 1995, when Netscape was the main browser vendor. As a result, he created JavaScript, a computer language that, when combined with HTML, adds dynamic capability to web sites. For instance, JavaScript helps administrators to modify the information on web website when such a situation happens, such as when a consumer selects a box. This even enable users to obtain and analyse user interaction. In May 1995, Eich implemented the JavaScript programming language in about ten days, a truly astounding achievement. One of Netscape's creators, Marc Andreessen, gave the new language the names Mocha and then LiveScript. But Andreessen was set on using the name JavaScript for marketing purposes. The software industry was enthralled by Java, a hot new programming language at the moment. Andreessen reasoned that if their new browser programming language included the name Java in it, all the Java fans would flock to it. Andreessen got his aim in December 1995, when Netscape received a trademark license from Java maker Sun Microsystems, and the name of LiveScript was changed to JavaScript. Unfortunately, many people have mistakenly assumed that JavaScript is the same as or quite similar to Java over the years. Don't let the name mislead you: JavaScript is nothing like Java. C++ and other popular programming languages are actually more similar to Java than JavaScript. To boost JavaScript's effect on all browsers (not only Netscape's browser), Netscape submitted it to the Ecma International1 standards body in 1996. The ECMAScript standard was developed by Ecma International using JavaScript as a foundation. As expected, ECMAScript has become the de facto standard for interactive programming languages in today's most popular browsers. The most recent ECMAScript version at the time of publication is version 7, which was released in 2016. It was challenging to come up with the term ECMAScript because different browser manufacturers had strong opposing viewpoints. The outcome, ECMAScript, according to JavaScript author Brendan Eich, is "an undesired trade name that sounds like a skin illness." Netscape founded the Mozilla free-software community in 1998, which went on to create Firefox, one of today's most popular browsers. Brendan Eich moved to Mozilla, where he and others have continued to improve JavaScript throughout the years, according to the Ecma International's ECMAScript standard. Other browser vendors have their own JavaScript implementations. Microsoft employs JScript in their Internet Explorer and Edge browsers. Google employs the V8 JavaScript Engine in its Chrome browser. Fortunately, all browser manufacturers aim to adhere to the ECMAScript standard, so programmers can write one version of their code that will function across all browsers for the majority of tasks. We use standard ECMAScript code in this book, which is compatible with all browsers. We refer to our code as JavaScript, as does practically everyone else in the online programming community, despite the fact that JavaScript is merely one of numerous ECMAScript implementations (JavaScript is the one used in Mozilla's Firefox). /8/

**2.2.** **CSS frameworks and preprocessors**

HTML5 and CSS3 are revolutionizing the way web pages are created. CSS3 introduced advanced capabilities like gradients, transitions, and animations to site designers. These new features, on the other hand, increased CSS code complexity, making it more difficult to maintain. Aside from the added complexity given by CSS3, writing CSS can become tedious over time because programmers must repeat many of the same tasks (such as looking up color values in CSS and declaring margins and padding). The inefficiency of these little, repetitive processes adds up quickly. Preprocessors solve these and a few other inefficiencies. /18/

CSS preprocessors add new features to CSS by incorporating notions from modern programming languages. It is mandatory to know how to manusript via CSS before you can utilize Sass (Syntactically Awesome Stylesheets). Datas, functionalities, operators can all be used with CSS preprocessors. The "Don't Repeat Yourself" (DRY) philosophy can be applied to CSS code with CSS preprocessors. You can avoid code repetition by using the DRY concept.

**What Are Preprocessors?**

A preprocessor transforms one type of data into another. Less and Sass are popular preprocessor languages in the CSS world, and typically groups prior with the Less or SCSS formats and output processed CSS. These CSS preprocessors make CSS more powerful by eliminating inefficiencies and making web site development easier and more rational. As preprocessors became more popular, several frameworks based on them arose; one of the most popular is Compass. CSS can be structured similarly to other languages like PHP or JavaScript using a preprocessor. As a result, a preprocessor gives the developer piece of mind. It allows you to develop future-proof and maintainable code, saving you time and effort. Preprocessors are CSS extensions, therefore legitimate CSS code is also valid preprocessor code. Any preprocessor will be easy for developers who are already familiar with CSS. /19/

**Preprocessors' benefits**

CSS is a declarative programming language. This means that the browser uses the styles you write in the code without having to compile them. Many programmers prefer to hand-write stylesheets. Preprocessors, they say, can also able to manipulate more and more confusing into its work mechanism as well as learning curve. CSS preprocessors, on the other hand, make your day-to-day work a lot easier. When it comes to knowing how different properties function, cascading, browser support for specific properties, selectors, quirks, and so on, CSS, which seems to be the cornerstone of the all the user - defined functions, has a high learning curve. Stylesheets, for the most part, are extremely repetitive, with properties or groups of properties, and so on. A linear document is the most common type of CSS file. This is enough to drive an object-oriented coder insane. According to the DRY principle, every piece of knowledge in a system should have a single, clear, authoritative representation. /20/



Figure 2.3 Comparison between SCSS and CSS code

The most basic rationale is that redundancy in code can lead to failure and misunderstanding among engineers. The DRY principle is not followed by CSS. CSS files frequently contain repeating rules, declarations, and values. Throughout their stylesheets, developers are continuously writing the same pieces of code. Parameters, metaphorical integers, regular expressions, procedures across parameters, and other programming language features are not available in CSS. Between your preprocessor-friendly file and the produced CSS files that will be sent to the browser is the CSS preprocessor. CSS preprocessors help you develop code that follows the DRY principle, making it faster, more efficient, and easier to maintain. Using the command line or an application, the code is subsequently compiled into ordinary CSS files. /9/

**SASS**

The most well-known preprocessor is Sass (Syntactically Awesome Stylesheets), which has been around for eight years. Hampton Catlin and Natalie Weizenbaum created and developed Sass, a preprocessor language. When SassScript is produced, CSS rules for various selectors supplied in the Sass file are generated. Sass can keep track of.sass and.scss items as well as construct the output.css version that when one of either being committed. Sass is a Ruby-based open source project. /21/

**LESS**

Less seems to be a prominent CSS generator, analogous as Sass. Mixins, aliases, etc stacked displayed being considered to enhance this same conventional CSS semantics. That's easy to operate. And via citing a Plugin to HTML content. Along with its practicality, Less has drawn the huge fan base. Less seems to be a free software project. That one was done via Ruby also in outset, but recent incarnations got implemented as JavaScript. Considering Less and CSS have about the same syntax, a lawful CSS program would also be a legitimate Less text. Less extends from these other CSS parsers in how it permits enabling authentic translation.

**Compass**

This script seems to be a CSS writing framework that is free and open source. Because Compass is based on Sass, it can take advantage of all of Sass' features. It is really popular and is currently being developed. Developers can use Compass to write better HTML without using presentational classes. Compass contains a large number of reusable patterns that are frequently used by developers. Bourbon, which is designed with Sass and for Sass by Thoughtbot, runs alongside Compass. Bourbon is a Sass mixin library that is simple and lightweight. The following sections go over some of Compass' most important features. /22/

**Bootstrap**

The Bootstrap framework is a set of tools for building webpages and web apps. Everything just incorporates HTML as well as CSS interface patterns for typography, fields, labels, navigational, and several other application programs, and also some JavaScript enhancements Twitter created Bootstrap as a framework to promote consistency among things required. Bootstrap provides available on all major platforms and features dynamic content, which indicates that only the structure of web pages alters dynamically based on the device being used (PC, tablet, mobile phone).

**2.3.** **JavaScript Workflow**

JavaScript is a programming language for turning web pages into apps. JavaScript allows users to interact with web pages without having to reload them.

**Why Would You Want to Use JavaScript?**

The browser becomes a platform for building applications thanks to JavaScript, which transforms it from a document viewer to a platform for developing apps. It's useful because it allows developers to edit the contents of a web page after it's loaded, allowing them to provide users immediate feedback when they alter a form, load resources on demand for performance reasons, or create full-fledged user interfaces that seem like desktop applications. Because JavaScript is so tightly connected with the browser, it's quite beneficial. This connection enables programmers to control multiple levels including its device's performance and perhaps even the entities on either the display. The execution model used by JavaScript is known as event-driven. While users incorporate JavaScript script inside a browser window, it just isn't performed until after the incident it is really associated to occurred. Loading the page, exiting the page, interacting with a form element in any way, clicking a link, or even scrolling up or down are all examples of events that can call JavaScript. There are plenty of other events to choose from as well. These events are frequently used in ways that most users would find irritating. When you visit one of their pages, many websites, for example, launch an other window with an advertisement. The page load event and JavaScript are used to do this. Other sites use JavaScript, which is triggered by the page unload event, to open additional windows after you leave them. Validating forms before they are submitted or presenting extra information on a page when a user clicks a link without requiring a complete page refresh are less obnoxious applications. JavaScript allows you to manipulate web pages without having to send a request back to the server, or to submit a request to the server to retrieve information without leaving the user's current page. You can utilize scripts embedded into your web pages to change the contents of a page, the style of items on a page, evaluate user input before a user submits a form, and affect browser behavior. /10/

**Javascript Engine**

A JavaScript engine, often known as a JavaScript interpreter, is a computer program or interpreter that runs JavaScript code. JavaScript engines can be written in a number of different languages. The V8 engine in Chrome browsers, for example, was created in C++, but the SpiderMonkey engine in Firefox browsers was created in C and C++. A JavaScript engine can take the shape of a regular interpreter or a just-in-time compiler that converts JavaScript to bytecode. Although the early JavaScript engines were nearly entirely interpreters, today's machines incorporate just-in-time (JIT) implementation help increase efficiency.

A JavaScript engine has been embedded into each and every contemporary browser. Followings are amongst the most extensively utilised JavaScript engines:

* Chrome’s V8 engine
* Firefox’s SpiderMonkey
* Safari’s JavaScriptCore (a.k.a Nitro, SquirrelFish and SquirrelFish Extreme)
* Edge’s Chakra

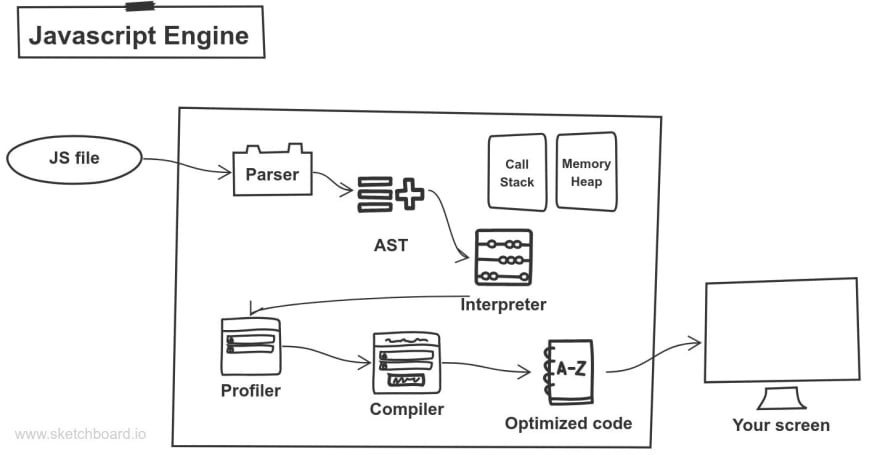


Figure 2.4 Comparison between SCSS and CSS code

**ECMAScript**

ECMAScript has a subset called JavaScript. At its foundation, JavaScript is ECMAScript, although it extends it. ActionScript, JavaScript, and JScript are all ECMAScript-based languages. As an example, AS/JS/JScript are three separate automobiles, but they all share the same engine... each of their exteriors, however, is distinctive, and each has undergone various modifications to make it such.

Brendan Eich created Mocha, which evolved into LiveScript, and then JavaScript. Netscape delivered JavaScript to Ecma International, a standard-setting organization, and it was called ECMA-262, or ECMAScript. /11/

**Different browser performance**

The command to "jump" will be understood by two persons, but one of them will react first since he was able to comprehend and process the command faster. Similarly, two browsers can read JavaScript code, but one is faster due to its more efficient JavaScript engine.

**Different browser support**

Someone in a big community of English speakers may be familiar with terms, idioms, and grammatical rules that others are unfamiliar with, and vice versa. The same is true with browsers. While all JavaScript engines in browsers understand JavaScript, some do it better than others. Browsers support the language in different ways. Although JavaScript engines interpret and execute JavaScript in detail, browser support is commonly referred to as "ECMAScript compatibility" rather than "JavaScript compatibility." /12/

**JavaScript runtime**

The JavaScript engine executes and interprets JavaScript code in this environment. JavaScript can run on and with host objects provided by the runtime. The “existing object or system” referenced in the scripting language definition is the JavaScript runtime. The code is parsed and parsed by a JavaScript engine, which then conducts the interpreted actions after an object or system parses the code and parses its job. The dog is walking, the person is running, and the figure in the video game is jumping. Because they give "host objects" at runtime, JavaScript programs can access apps. The JavaScript runtime is the web browser on the client side, whenever host objects such as windows and HTML files seems to be made accesible for implementation. Basic JavaScript does not include such objects. They're called web APIs since they're objects exposed by a browser that acts as a JavaScript host environment. Node.js is the JavaScript runtime on the server. The file system, processes, and requests are just a few of the server-related host objects provided by Node.js. Surprisingly, the same JavaScript engine can be used by many JavaScript runtimes. V8 is an engine that is used in two very distinct environments: Google Chrome and Node.js, for example. /13/

**2.4. JavaScript most popular frameworks**

**JavaScript libraries**

JavaScript has the most comprehensive collection of actually useful libraries and frameworks of either of the technologies that would be utilised when generate web apps. The main distinction among such kind of plugins which were being made up of characteristics that only the machine that can be used to implement a project, whereas frameworks specify how an application is designed. In other words, rather than the other way around, the framework calls on the application code. Of course, libraries are still required for developers to complete fundamental web operations with JavaScript. In recent years, the nature of these technologies has shifted. For example, in 2015, jQuery was the most popular JavaScript library. Traditional JavaScript libraries are still relevant today, according to JavaScript Scene, but they are seeing greater competition from JavaScript frameworks like React, Angular, Node.js, and Ember.js. /14/

**D3.js**

Many current websites rely heavily on data. For example, news sites must continually update their material, making it impractical to adjust the system every few seconds to accommodate these changes. D3.js, or Data Driven Documents, is a unique package in that it prioritizes data. The key benefit of the whole module is that it is the capacity to maintain content in a very flexible manner. D3.js' comparatively simple presentation, in addition to its enormous versatility, helps increase speed by avoiding a lot of the baggage that other libraries utilize to make bedazzling displays. If D3.js is still a little too much for your development needs, Chart.js is a simpler package specialized to delivering charting. /15/

**jQuery**

jQuery has established itself as the undisputed king of the web page. For three important reasons, many websites still utilize jQuery for core DOM (DOM) manipulation:

* jQuery is quite simple to learn. It has a lot more examples than other libraries, so there's a strong chance you'll discover one that demonstrates how to complete your unique task.
* jQuery is lightning quick. jQuery makes it easy to execute specific activities quickly since it focuses on doing them well.
* jQuery has a big user base. It also has a large number of supporters and contributors from the community.

**React**

React is occasionally referred to as a framework, despite the fact that it is usually regarded a library. React provides the display element of the Model-View-Controller (MVC) paradigm to building large-scale applications. It doesn't make any assumptions about the technological stack you're using to model or handle your data. React is only concerned with showing data on the screen. With this in mind, it avoids using the browser's Document Object Model (DOM) in favor of a lightweight virtual DOM that displays data quickly. One of the most appealing aspects of React is that it is declarative, which also means code owner is able to order the framework what you want to accomplish rather than how to do it.

**JavaScript Frameworks**

**Angular**

By functioning as the controller portion of MVC, the Angular framework allows you to expand HTML. The behavior of DOM components is controlled by controllers. Angular allows for the creation of new behaviors in a simple and clear manner. The extensions end up looking more like HTML extensions than something tacked on. The Angular website emphasizes two key reasons to utilize this product: "speed and performance" and "amazing tooling." The coding, on the other hand, can grow complicated, and this framework may be better suited to large corporations than small businesses. Even the simplest Angular examples rely on numerous files, which quickly add complexity. Recent edition of Angular uses TypeScript rather than pure JavaScript, which adds to the learning curve but improves scalability. Fortunately, Angular comes with a comprehensive tutorial, which can be found here, that should help experienced developers get up and running quickly.

**Node.js**

Node.js is a JavaScript runtime that is asynchronous and event-driven. It has certain unique features. For one thing, if it doesn't have any work to perform, it just "sleeps." Furthermore, because Node.js does not require locks to function, it does not have the same tendency as other frameworks to deadlock. This product is more like jQuery than jQueryUI in the sense that it isn't used to construct a user interface.

Instead, developers execute commands which really reacts to events—events are generated by that of the customer, and indeed the receiver uses for them. Any result is displayed via the user interface, which is a separate element.

**Vue.js**

Vue is a versatile open-source JavaScript framework that is "progressive," meaning that it would be "conceived from either the cradle to the grave to be steadily reusable," unlike "monolithic frameworks." Vue uses a "component-based development methodology" that allows you to customise Vue modules throughout existing applications, tend to range from just a plugin to a full-featured framework. Elements, layouts, transitions, and two-way data binding are all important parts of Vue, but its "reactivity" approach is likely its most distinguishing characteristic. Reactivity simply implies that modifying a JavaScript object in Vue changes the Vue templates instantly and unobtrusively. /16/

**CHAPTER 3. Construction of online examination portal**

**3.1**. **Getting environment ready for the project**

**VS Code**

Visual Studio Code had already gained prominence from its very launching throughout 2015 as one of the greatest lightweight programming languages and code editors. It's simple to understand why VS Code is a favorite of many programmers and amateurs alike, with a variety of features that support a variety of languages and a big library of extensions to expand upon the basic text-editing functionalities.

**How to Create a New Project in VS Code**

VS Code does not use the standard “File > New Project” interface to begin working on new projects, as most IDEs do. VS Code's core architecture is that of a text editor that searches your file system for existing files to update and compile. As a result, it is unable to provide sufficient scaffolding for new construction projects on its own.

**Google Chrome**

Google Chrome empowers staff to collaborate as well as engage within various activities in some kind of a pleasurable yet productive way. Everything just includes the following prospects:

* That software seems to be uncomplicated to manage swift, as well as trustworthy.
* Constructing webpages for something like the future Internet.
* Utilising this same greatest up-to-date APIs across online platforms.

**3.2**. **Creating frontend folder and file structure**

To proceed with both the supported languages regarding Software development as well as UI scripting:

HTML5, CSS3, JavaScript/jQuery, Bootstrap and SCSS are almost all implementations of markup languages.

HTML5 is a markup language that creates the web page's fundamental skeleton. Cascading Style Sheets (CSS3) are used to style HTML. JavaScript is a scripting language that allows you to add functionality and interactions to your website. Bootstrap is a framework for creating mobile-first, responsive websites. SCSS, also known as SASS, is a CSS pre-processor that adds new functionality to CSS.

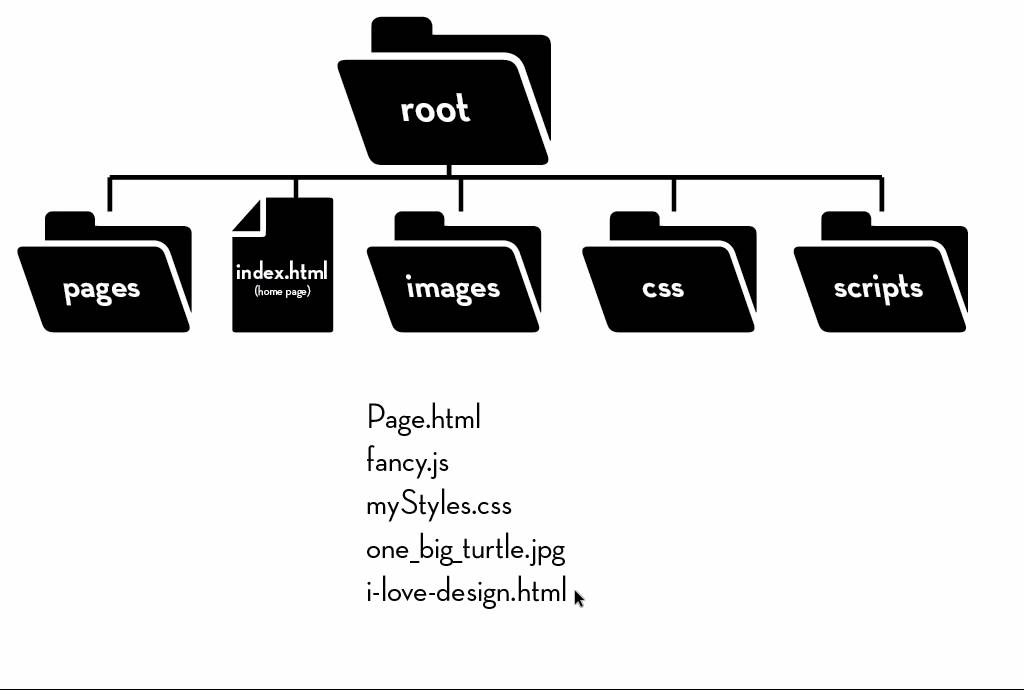


Figure 3.1 Folder infrastructure

The first is the project's primary Root folder, which contains all of the project's files and directories as well as the project's primary web pages. Most of the more significant things were just as shown in:

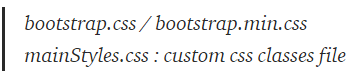


Figure 3.2 Design and stylesheet files

The JavaScript files should be under the js folder:

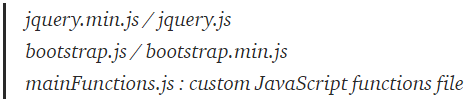


Figure 3.3 JavaScript script files

All of the photos and media assets will be stored in the “img” folder. In general, constructing sub-folders within the “img” folder to organize assets by sections and pages; for example, all favicons go in the favicon folder, and all home page images go in the home folder.

All foreign fonts will be stored in the “fonts” folder, which we will subsequently import into CSS and use to construct font families. The common folder, also known as the “components” folder, contains all of the common HTML files that will be imported into the major pages with JavaScript later.



Figure 3.4 Addressing into element for interaction

Every other section will also have an unfilled component with the property as well as identifier ‘nav container,' by using the regular expression, designers should import the navbar.html throughout the division. This same upside would be that if company want 15 pages, you only need to modify the navbar.html section, and indeed the tasks are completed. However, this only works if you run the code on a server, which one can do with Live Server for VS Code, which generates a local server, or users may use any other approach.

**3.3**. **Constructing Homepage**

The homepage is the initial section of the project that appears when you enter and update the web page. It provides general information as well as some routes that link to other side pages (components) and allow interaction with them. The primary web page of a website is called the home page (or homepage). When an application starts up, one or more pages are always displayed on the web browser. It's also known as the initial page in this scenario. The word "home" stems from the ability to return to the initial page at any moment by pressing the Home key on a keyboard.

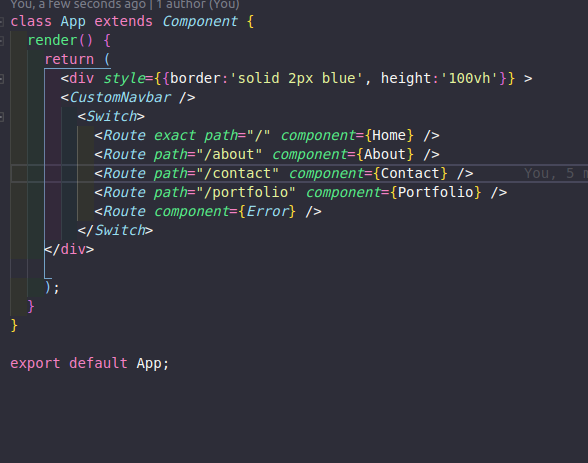


Figure 3.5 Main page’s layout

**Maintaining design**

It's critical to preserve the homepage's normative concept, layout and wasted space. Because once customers being overwhelmed with far too much content and imagery, they becomes exhausted. If indeed the information in company webpage is well-thought-out and orderly, users could seem greater disciplined and knowledgeable.

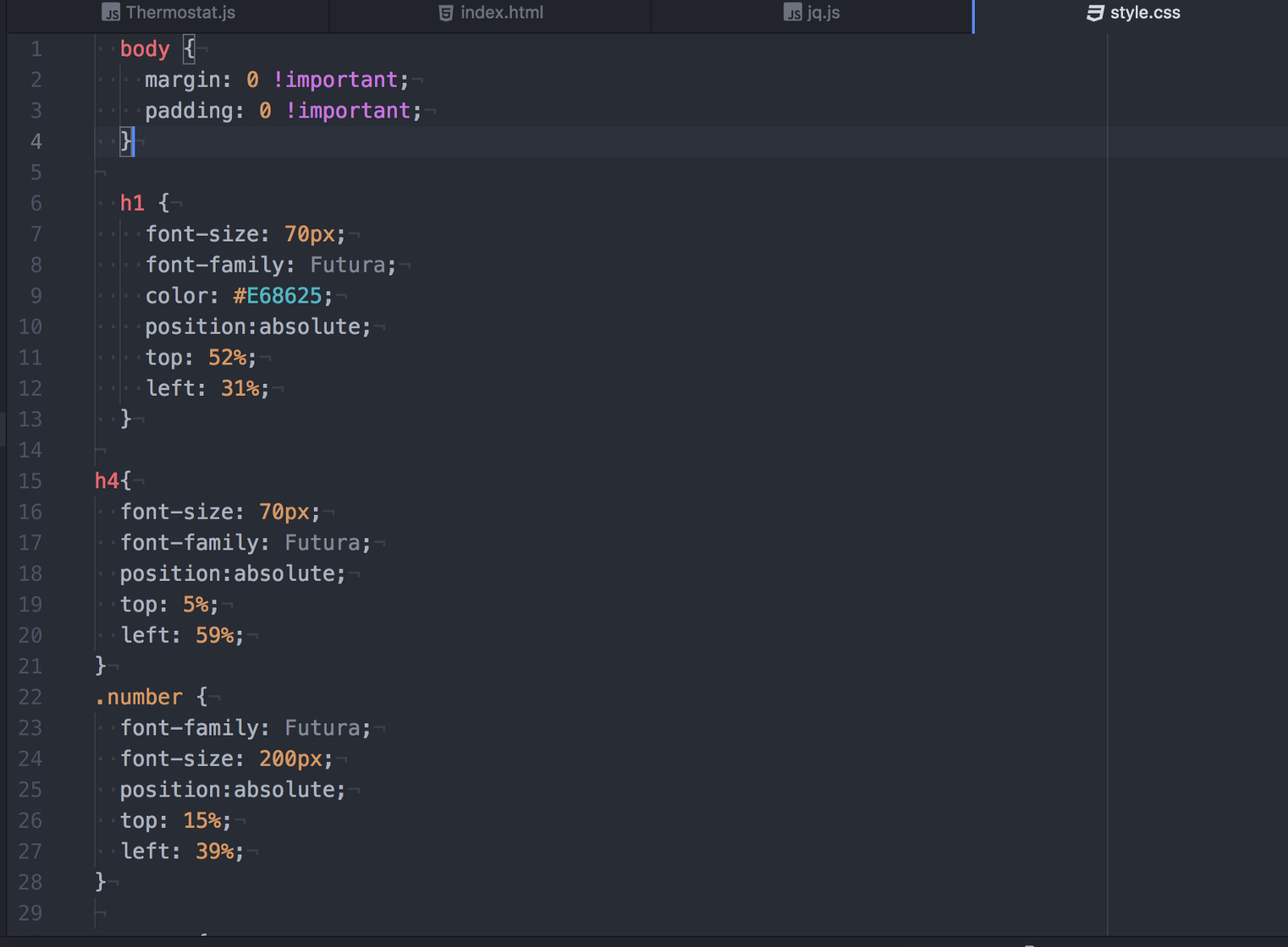


Figure 3.6 CSS code for Homepage

Iterators are useful because they allow you to access individual items in a collection while keeping track of the current index. We pass in a basic array with two values and call it to iterate over the items.

following ().



Figure 3.7 JavaScript code for generating new array elements

**Applying Bootstrap4 design**

Bootstrap is being used to interact with design codes in order to make them more useful. Because of Bootstrap has shortcases to make writing codes more easy, it enables to build projects in less time and more efficiency

As shown below, some Bootstrap codes was used in order to design elements in project. Bootstrap seems to be a fair and democratic Development methodology about front project management that really is cross - browser compatible. It requires various patterns for typographic, fields, and other application programs which are already predicated on CSS as well as JavaScript.

* Complete redesign of the code 2. Substituting Sass for Less
* Adding the ability to customize the navigation
* Implementing responsive spacing and sizing tools

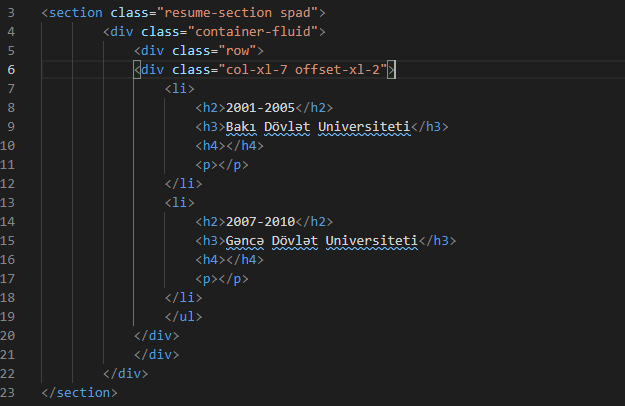


Figure 3.8 Applying Bootstrap to design

**3.4**. **Adding plugins to the project**

It's quick, light, and makes navigating HTML documents and manipulating page elements a breeze. Many jQuery plugins have been written on top of the framework to add functionality to webpages due to its extensibility. There is a jQuery plugin for almost any piece of functionality that your website requires.

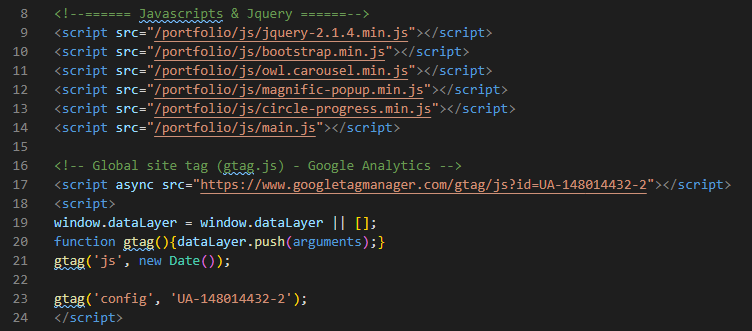


Figure 3.9 Importing plugins

**CONCLUSION**

Test system was intended to be used in mock exams conducted by other institutions with minor modifications. The created method generates a list of any student's provided answers for each examination. This initiative has provided a diverse range of study opportunities. The theoretical understanding of the digital education system has reinforced the foundation for moving on with the project. The project's practical implementation phase has improved web development capabilities. It has also provided the opportunity to observe the security measures. During the portal's implementation, a standard security framework was followed. The potential weaknesses have been discovered, and the suggested measures for overcoming those hurdles have been assessed.

**REFERENCES**

1. Front-End Web Development: The Big Nerd Ranch Guide by Chris Aquino and Todd Gandee
2. HTML & CSS by Jon Duckett
3. HTML in easy steps by Mike McGrath
4. HTML & CSS: Design and Building Web Sites by Jon Duckett
5. HTML & CSS and JavaScript web publishing by Laura Lemay
6. Web programming with HTML,CSS,JS by John Dean
7. HTML, CSS, and JavaScript All in One by Jennifer Kyrnin and Julie Meloni
8. Front-End Tooling with Gulp, Bower, and Yeoman by Stefan Baumgartner
9. Beginning CSS Preprocessors by Anirudh Prabhu
10. JavaScript: The Good Parts 1st edition by Douglas Crockford
11. Secrets of the JavaScript Ninja by John Resig, Bear Bibeault, Josip Maras
12. Clean Code: A handbook of Agile Software Craftsmanship by Robert C. Martin
13. JavaScript and JQuery: Interactive Front-End Web Development by Jon Duckett
14. Sass and Compass in Action by Wynn Netherland, Nathan Weizenbaum, Chris Eppstein, Brandon Mathis
15. High Performance Web Sites by Steve Souders
16. <https://developer.mozilla.org/en-US/docs/Web/HTTP>
17. [https://www.geeksforgeeks.org/web-development/?ref=shm](https://html.com/#HTML_Reference_Guides)
18. <https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/The_head_metadata_in_HTML>
19. [https://www.w3schools.com/js/js\_statements.asp](https://html.com/#HTML_Reference_Guides)
20. <https://html.com/#HTML_Reference_Guides>
21. <https://www.tutorialspoint.com/javascript/javascript_syntax.htm>
22. <https://www.tutorialspoint.com/javascript/javascript_functions.htm>